

Laboratorio de Computación II
TRABAJO PRÁCTICO N° 3 - Parte I

Presentación del trabajo

La aplicación sirve para obtener información sobre la producción de un establecimiento ganadero, donde se produce leche y se presta el servicio de engorde.

⑩ Sección “Existencias ganaderas”

En la sección denominada “Existencias ganaderas” se puede consultar el listado completo de bovinos, con su identificación única, uso (tambo o engorde), fecha de ingreso, sexo y raza.

La lista de bovinos se carga desde un archivo json denominado StockGanadero.json y ubicado en la carpeta Archivos dentro del proyecto.

El usuario puede agregar bovinos (a menos que ya haya un bovino cargado con el mismo identificador, que debe ser único), eliminarlos o modificar sus datos (excepto el identificador único). Tampoco se pueden agregar bovinos machos destinados a la actividad de tambo.

Al presionar el botón volver se vuelven a guardar en el archivo StockGanadero.json y se da un aviso al usuario si todo salió OK.

Asimismo, el usuario puede exportar los datos, tanto en formato .json, como .xml y en .txt.

⑩ Sección “Información productiva”

En la sección denominada “Existencias ganaderas” se puede visualizar la siguiente información y estadísticas sobre el establecimiento:

Datos sobre el stock ganadero:

- Distribución total por sexo
- Distribución total por uso

Datos sobre la producción lechera:

- Cabezas destinadas al tambo
- Producción de leche en el último año: Se obtiene un promedio diario para cada bovino destinado a tambo de manera aleatoria (entre 20 y 50 litros diarios). Luego se calculan los últimos 365 días de permanencia para los animales que tienen mas de 1 año en los establecimientos, o todos los días para los que tienen menos de un año. De esta manera, se obtiene la producción total en el último año calendario.
- Raza más productiva para tambo.

Datos sobre la producción ganadera:

- Cabezas destinadas al engorde

- Producción de leche en el último año: Se obtiene un promedio diario para cada bovino destinado a engorde de manera aleatoria (entre 0 y 2 kilos diarios). Luego se calculan los últimos 365 días de permanencia para los animales que tienen mas de 1 año en los establecimientos, o todos los días para los que tienen menos de un año. De esta manera, se obtiene la producción total en el último año calendario.
- Raza más productiva para engorde.

Además, se puede exportar la información obtenida tanto a xml, como a json y a txt.

Aplicación de temas aprendidos (clases 10 a 15)

⑩ Clase 10: Excepciones

La aplicación valida datos a fin de evitar que se generen excepciones y el programa se cierre abruptamente, sin poder brindar una explicación al usuario.

Para los casos en los que es más probable que se generen excepciones, se realizó el correspondiente manejo de las mismas mediante bloques try catch.

En el caso de manejo de archivos, siempre se realiza un bloque catch para excepciones de tipo FileNotFoundException y DirectoryNotFoundException que lanza una excepción con un mensaje afín y más humano a fin de atraparlo desde el Program.cs o el Form que se esté utilizando. Al exportar datos, como la ruta de la archivo de destino es un dato ingresado por el usuario tambien se valida que dicho campo no este vacío.

Además, se crearon una excepción propia AnimalExistenteException, la cual se lanza en el caso de que el bovino que se quiere agregar a la lista ya se encuentre en la misma (tenga el mismo identificador).

⑩ Clase 11: Pruebas unitarias

Se cuenta con un proyecto de Test Unitarios, en el cual se realizan múltiples pruebas a los diferentes métodos de la biblioteca de clases, la cuales esperan tanto valores de variables como de excepciones.

Ejemplos:

CalcularPermanenciaEnDias_RecibeAnimalIngresadoHoy_DeberiaDevolver1()

AgregarAnimal_RecibeAnimalQueYaEstaEnLaLista_DeberiaDevolverAnimalExistenteException()

⑩ Clase 12: Tipos genéricos

Se cuenta con una clase estática Archivos que recibe un tipo de referencia, al cual se va a serializar a un archivo que puede ser xml, json o txt. A ese tipo tambien se deserializa desde json.

⑩ Clase 13: Interfaces

Se crea la interfaz IBovino, que define los métodos que deben ser implementados para estos animales. Esta es utilizada por la clase Bovino.

⑩ Clase 14: Sistemas de archivos

Se utilizan varios métodos de las clases utilizadas para manejar el sistema de archivos.

Ejemplos: Para obtener el directorio actual al momento de definir el path se utiliza `Directory.GetCurrentDirectory()` , para armar una ruta se utiliza `Path.Combine()` y para verificar si un archivo existe se utiliza `File.Exists()`.

⑩ Clase 15: Serialización

La aplicación cuenta con persistencia de datos, el listado de bovinos se obtiene de un archivo formato json que se encuentra en la carpeta del proyecto.

Tanto el listado de bovinos como la información obtenida del análisis de datos se pueden exportar en los formatos json, xml y txt, utilizando los métodos de la clase genérica `Archivos.cs`.